Design and synthesis of robust controllers for hybrid systems using modal logic: Part II

(extended abstract)

J.M. Davoren and T. Moor
Research School of Information Sciences and Engineering
Australian National University
Canberra ACT 0200
Australia

E-mail: davoren@arp.anu.edu.au, tmoor@syseng.anu.edu.au

The logic-based synthesis algorithm developed in this work is an elaboration of, and a significant advance on, a simpler procedure first sketched in [7]. The essential idea is that in designing and constructing a hybrid controller for a given plant and given specifications, one needs to reason about *sets of plant states*, and build up more complicated sets of states by applying various operators arising from the flows and the specification data. Following [7], we use *modal logic* as a clean and elegant formalism in which to conduct such reasoning about sets of states, and to custom-design operators on sets tailored to the specifications. The logic gives us the technical tools with which to formulate a general and finitely terminating synthesis algorithm which applies uniformly to arbitrary differential equations $\dot{x} = F_c(x)$, subject only to standard assumptions on the existence and uniqueness of solutions, with finite termination analytically derived from an assumption of compactness. By formulating these constructions of complex sets of states in the language of modal logic, we gain the immediate pay-off that the *correctness* of the synthesis procedure – that any controller generated by the procedure does indeed ensure that the closed-loop system satisfies all the performance specifications – can transparently be shown to be a *formal deductive consequence* of a theory of modal formulas that are true of that hybrid automaton model purely in virtue of the construction.

As an illustrative example, we consider a plant in $\mathbb{R}^2$ given by three linear differential equations: the flows of two of them are stable spirals, and the other flows in horizontal straight lines. For the safety specification, the Bad region is a disk of radius 2 centred at the origin, together with the complement of the box $[-10, 10] \times [-10, 10]$. The two spirals have their centres lying on the boundary of this box, at the points $(-10, 0)$ and at

$(10, 0)$ respectively. The prescribed event sequence behaviour is to proceed with a clockwise motion, navigating through the partition blocks consisting of the four quadrants intersected with the complement of *Bad*. Visually, we are steering a point $x \in \mathbb{R}^2$ clockwise around the disk and within a bounded box, using switching sequences chosen from three primitive control actions: a *downwards* motion from the left spiral, an *upwards* motion from the right spiral, and an *across* motion, to the right above the horizontal axis and to the left below that axis. The three flows and the safety and event sequence specifications for this example are illustrated in Figure 3.

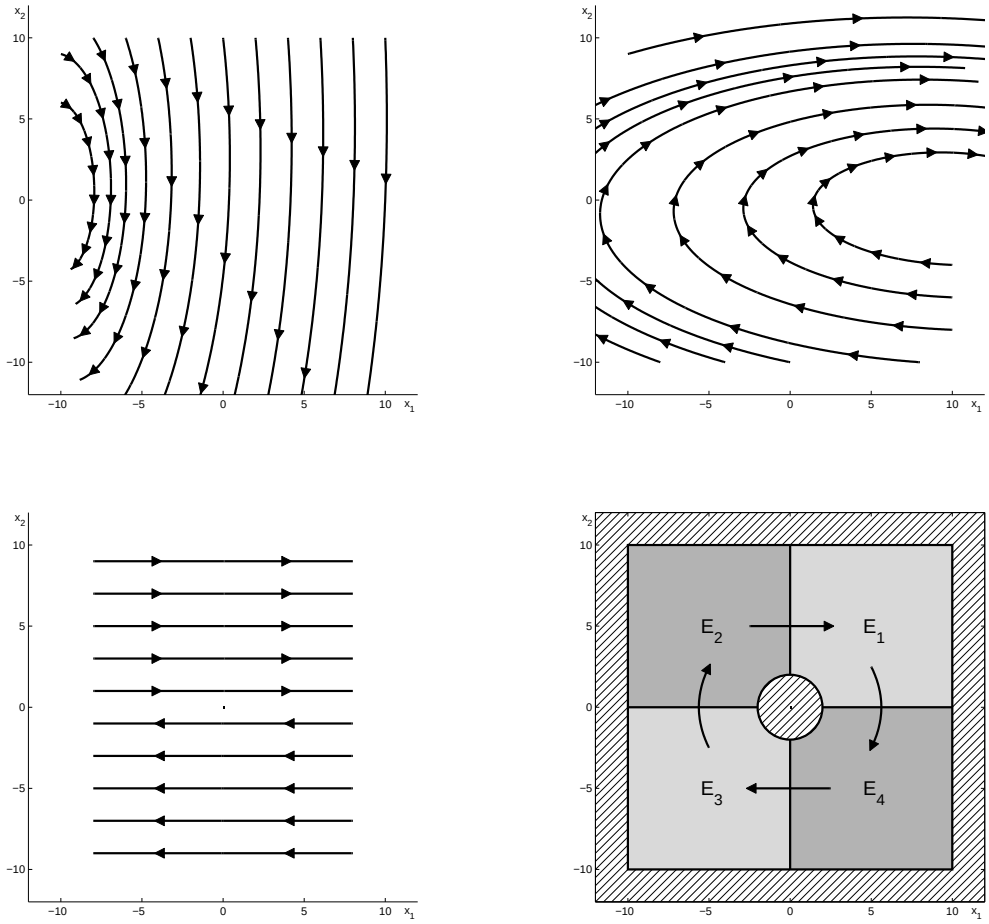

Figure 1: Three flows in $\mathbb{R}^2$, and safety and event sequence specification sets

Modal logic provides a general-purpose framework in which to formalize a wide variety of operators on sets. In addition to the reachability operators of flows of differential equations, we also consider the operator which, given

a set A, returns the *δ-expansion* of, or *δ-ball* around A, for some fixed metric distance $\delta > 0$; by this we mean the set of points lying within distance δ of some point in A. The dual notion is the *δ-contraction* of A, meaning the set of points in A around which one can fit a ball of radius δ that still lies wholly inside A. By using these notions of metric tolerance, we are able to cleanly formulate and prove a form of *robustness* or *tolerance property* for our synthesis algorithm. Our result is that not only is it the case that all hybrid trajectories of the resulting closed-loop system H meet the given specifications, but in addition, all hybrid trajectories arising from certain *bounded variations* of H will still meet those specifications. The variation classes we consider arise by allowing a bounded degree of tolerance of sensor and actuator imprecision, as well as bounded variations in the differential equations defining the plant, where the variation depends continuously on a parameter. Both the latter, traditional formulation of robustness in terms of plant variation, as well as our notions of tolerance, fall within a larger framework of robustness concepts for hybrid automata proposed by Horn and Ramadge in [10].

The modal logic framework also gives us a clean way to separate out the determination of what sets need to be computed, and the structure and correctness of the abstract solution algorithm, from the distinct issue of how and when such an algorithm can be *effectively implemented*. Effective implementation requires a finitary symbolic means of representing set of states $A \subseteq \mathbb{R}^n$, with respect to which the Boolean and modal logic operators can be effectively evaluated, and furthermore, the representation of sets must be *decidable* in the sense that it can be determined by finite computation whether distinct representations are semantically equal (or equivalently, whether a representation is of the empty set). These are the fundamental issues for the application and development of symbolic model checking technology to hybrid and real-time systems [1, 2, 7, 9]. There are two main approaches to effective implementation, which we briefly outline here.

The first approach works with *exact symbolic representations* of state sets $A \subseteq \mathbb{R}^n$, whereby sets are explicitly defined by first-order logic formulas with variables ranging over the real numbers; for example,

$$A = \{\, (x_1, \cdots, x_n) \in \mathbb{R}^n \mid P(x_1, \cdots, x_n) > 0 \ \ \wedge \ R(x_1, \cdots, x_n) \neq 0 \,\}$$

where P and R are polynomials in n variables with rational coefficients. The application of modal operators translates into first-order logic as the build-up of formulas using nested quantifiers ($\exists$ and $\forall$), and the evaluation of the modal operators can be performed using *quantifier elimination* tools such as REDLOG [8] or QEPCAD – although the quantifier elimination algorithm

implemented in those tools has been significantly improved [3]. Using such tools, our synthesis algorithm can be effectively implemented by exact symbolic means when the specification data sets Bad and E_k are *semi-algebraic* (defined by Boolean combinations of polynomial inequalities), and the plant equations $\dot{x} = F_c(x)$ are defined by a nilpotent matrices, or otherwise have polynomial flows. The recent work of [12, 13] studies the exact symbolic computability of reachability operators for flows of a more general subclass of linear systems. Note, however, that our running example with spiral flows falls outside this class.

The second approach to effective implementation adopts the strategy of *approximated representation* of sets of states, working with under- or over-approximations rather than exact representations of sets. Under this approach, one first discretizes the plant state space $X \subseteq \mathbb{R}^n$ using a finite cell cover (or partition), e.g. from a finite rectangular grid of boxes on a bounded space, or using finitely many polyhedra or ellipsoids. For a given set $A \subseteq X$, an under-approximation can then be obtained by taking the union of all cells lying inside A, while an over-approximation comes from a union of all cells which intersect A. For each abstract operator Op on sets used within a synthesis construction or broader model checking framework, one must develop procedures which given as input a set A represented as a union of cells, return as output a union of cells which is an under- (or over-) approximation of the set $Op(A)$. Recent contributions to approximation methods for the basic forwards and/or backwards reachability operators of differential equations (and differential inclusions) are variously based on boxes [2, 4, 14, 15, 16], polyhedra [5] or ellipsoids [11]. Each of these approximation techniques apply to arbitrary linear differential equations, and in principle, any of them could serve as a basis for approximated versions of the modal operators used in our abstract synthesis algorithm. Our first prototype software implementation is based on a discretization using boxes, and we use its graphic output to illustrate the steps of the synthesis algorithm on our 2-D example.

At the workshop, this work will be presented in two 20 minute talks, one each by Davoren and Moor. This second part will present the mathematical tools of modal logic used in the synthesis algorithm, and our software implementation of the algorithm using an approximated representation of state sets.

A full paper has been submitted for publication, and is available as a technical report [6] from the web address: `http://arp.anu.edu.au/~davoren/hybrid_control/hybrid_control.html`. There are also links to some video files of closed-loop simulations of solution controllers for example plant models and specifications generated by our prototype software implementation.

References

[1] R. Alur, T.A. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88:971 – 984, July 2000.

[2] E. Asarin, O. Bournez, T. Dang, O. Maler, , and A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, 88:1011 – 1025, July 2000.

[3] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43:1002–1045, 1996.

[4] O. Bournez, O. Maler, and A. Pnueli. Orthogonal polyhedra: representation and computation. In F. Vaandrager and J. van Schuppen, editors, *Hybrid Systems: Computation and Control (HSCC'99)*, LNCS 1569, pages 46–60. Springer-Verlag, 1999.

[5] A. Chutinan and B.H. Krogh. Computing polyhedral approximations to flow pipes for dynamic systems. In *Proc. of the 37th International Conference on Decision and Control, CDC'98*. IEEE Press, 1998.

[6] J.M. Davoren and T. Moor. Logic-based design and synthesis of controllers for hybrid systems. Technical report, Dept. Systems Engineering, RSISE, ANU, July 2000. `http://arp.anu.edu.au/~davoren/hybrid_control/hybrid_control.html`.

[7] J.M. Davoren and A. Nerode. Logics for hybrid systems. *Proceedings of the IEEE*, 88:985 – 1010, July 2000.

[8] A. Dolzmann, T. Sturm, and V. Weispfenning. Real quantifier elimination in practice. In B. Matzat, G. Greuel, and G. Hiss, editors, *Algorithmic Algebra and Number Theory*, pages 221–248. Springer-Verlag, Berlin, 1998.

[9] T.A. Henzinger and R. Majumdar. A classification of symbolic transition systems. In *Proc. of 17th International Symposium on Theoretical Aspects of Computer Science (STACS'00)*, LNCS. Springer-Verlag, 2000.

[10] C. Horn and P.J. Ramadge. Robustness issues for hybrid systems. In *Proceedings of the 34th International Conference on Decision and Control, CDC'95*, pages 1467–1472. IEEE Press, 1995.

[11] A.B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In N. Lynch and B. Krogh, editors, *Hybrid Systems: Computation and Control (HSCC'00)*, LNCS 1790, pages 203–213. Springer-Verlag, 2000.

[12] G. Lafferriere, G.J. Pappas, G. Schneider, and S. Yovine. Symbolic reachability computation for families of linear vector fields. *Journal of Symbolic Computation*, 2000. to appear.

[13] G. Lafferriere, G.J. Pappas, and S. Yovine. A new class of decidable hybrid systems. In F.W. Vaandrager and J.H. van Schuppen, editors, *Hybrid Systems: Computation and Control (HSCC'99)*, LNCS 1569, pages 137–151. Springer-Verlag, 1999.

[14] O. Maler and T. Dang. Reachability analysis via face lifting. In T.A. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control (HSCC'98)*, LNCS 1386, pages 96–109. Springer-Verlag, 1998.

[15] T. Moor and J. Raisch. Discrete control of switched linear systems. In *Proceedings of the European Control Conference 1999*, 1999.

[16] T. Moor and J. Raisch. Approximation of multiple switched flow systems for the purpose of control synthesis. In *Proc. of the 39th International Conference on Decision and Control, CDC'00*. IEEE Press, 2000. Submitted.